



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorio de Computacion Salas A y B

Profesor(a):

Asignatura:

Grupo:

No de practica(s):

Integrante(s):

No de lista o brigada:

Semestre:

Fecha de entrega:

Observaciones:

Calificacion:

Índice

1. Introducción	2
1.1. Planteamiento del Problema	2
1.2. Motivación	2
1.3. Objetivos	2
2. Marco teórico	2
2.1. Factorial Recursivo	2
2.2. Secuencia de Fibonacci	2
2.3. Conjetura de Collatz	3
3. Desarrollo	3
3.1. Descripción de la Implementación	3
3.2. Pruebas Realizadas	3
4. Resultados	6
5. Conclusiones	7
Referencias	8

1. Introducción

1.1. Planteamiento del Problema

La práctica consistió en implementar en Java un programa con menú que resuelva tres ejercicios clásicos: factorial recursivo, serie de Fibonacci y conjetura de Collatz. Estos problemas permiten aplicar conceptos vistos en clase, mejorar el uso del lenguaje Java y mejorar la lógica de programación.

1.2. Motivación

Practicar con estos algoritmos fortalece el entendimiento de la recursividad, ciclos y estructuras básicas de Java. Además, integrar las opciones en un menú ayuda a hacer el programa más interactivo y probar diferentes casos en una misma ejecución.

1.3. Objetivos

- Implementar en Java los algoritmos de factorial, Fibonacci y Collatz.
- Diseñar un menú que permita ejecutar cada opción.
- Documentar la solución teórica, la implementación y los resultados obtenidos.

2. Marco teórico

2.1. Factorial Recursivo

La función factorial es una fórmula matemática representada por el signo de exclamación “!”. En la fórmula Factorial se deben multiplicar todos los números enteros y positivos que hay entre el número que aparece en la fórmula y el número 1.[1]

$$7! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 = 5040 \quad (1)$$

No obstante, se ha acordado que en el caso de 0 factorial el resultado será igual a 1:

$$0! = 1 \quad (2)$$

2.2. Secuencia de Fibonacci

La secuencia de Fibonacci es una sucesión infinita de números naturales, descrita por primera vez por el matemático italiano Fibonacci en el siglo XIII. Esta serie numérica empieza con 0 y 1, siguiendo con la suma de los dos números anteriores hasta el infinito: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233.

La secuencia de Fibonacci forma parte de numerosos sistemas, incluso de la naturaleza, por ejemplo, se encuentra en la disposición de las ramas de los árboles, los pétalos de las flores y las hojas de un tallo, siendo el girasol el ejemplo más preciso. Cada uno de los números de Fibonacci se acerca a la llamada proporción áurea, proporción dorada o número de oro, lo que representa la perfección geométrica natural.[2]

2.3. Conjetura de Collatz

La conjetura de Collatz fue enunciada por el matemático alemán Lothar Collatz y para enunciarla necesitamos definir la secuencia de Collatz de un número. Tomemos un número natural cualquiera, el 2, el 54, el 17, el 20983, ... si dicho número es par, lo dividimos entre dos, y si es impar lo multiplicamos por 3 y le sumamos 1. Al reiterar este proceso obtenemos la secuencia de Collatz de dicho número. Por ejemplo, tomemos el número 13, teniendo en cuenta lo anterior, su secuencia de Collatz es la siguiente: [3]

$$13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow \dots \quad (3)$$

3. Desarrollo

3.1. Descripción de la Implementación

Para la practica se desarrollaron tres funciones principales: cálculo de factorial de manera recursiva, generación de la sucesión de Fibonacci de forma recursiva, y la aplicación de la conjetura de Collatz para un número dado.

El programa cuenta con un menú de opciones mostrado al usuario en consola, el cual permite seleccionar qué operación ejecutar. La interacción se realiza por medio de la clase Scanner, que recibe los valores ingresados por teclado. Una vez elegido el método, el programa ejecuta la función correspondiente y devuelve el resultado en pantalla.

- Factorial recursivo:

El método del factorial recursivo aplica la definición matemática de la función factorial, utilizando un caso base y llamadas sucesivas al mismo método hasta llegar a la condición de terminación.

- Fibonacci recursivo:

El método de la sucesión de Fibonacci recursiva se diseñó para calcular los términos de la serie aplicando la suma de los dos valores anteriores, con casos base para los primeros números de la secuencia.

- Conjetura de Collatz:

El método de la conjetura de Collatz implementa un ciclo que evalúa si el número ingresado es par o impar, aplicando las reglas establecidas (dividir entre dos o multiplicar por tres y sumar uno) hasta llegar al valor de 1.

La implementación del menú se apoyó en estructuras de control como condicionales y bucles, mientras que para la entrada de datos se utilizó la clase Scanner, lo cual permitió capturar valores desde el teclado. Finalmente, los resultados fueron desplegados en la consola, mostrando paso a paso el proceso de cada algoritmo.

3.2. Pruebas Realizadas

Factorial recursivo:

Entrada: 6

Salida esperada: 720

```

PS C:\Users\52551\Programacion\JavaDeidad> javac MenuRecursivo.java
PS C:\Users\52551\Programacion\JavaDeidad> java MenuRecursivo

=== MENU DE OPERADORES RECURSIVOS ===
1. Factorial
2. Fibonacci
3. Collatz
4. Salir
Elige una opcion: 1
Ingresa un numero para calcular su factorial: 6
6! = 720

```

Primer se compila con javac y el nombre del archivo, para posteriormente mandarla a llamar, que al elegir la opción 1 despliega la opción de ingresar el numero a calcular su factorial.

Fibonacci recursivo:

Entrada: 7

Salida esperada: 0, 1, 1, 2, 3, 5, 8

```

=== MENU DE OPERADORES RECURSIVOS ===
1. Factorial
2. Fibonacci
3. Collatz
4. Salir
Elige una opcion: 2
?Cuantos terminos de Fibonacci deseas ver?: 7
Sucesion Fibonacci: 0 1 1 2 3 5 8

```

Al elegir la opción 2 se despliega la opción para ingresar el numero de términos que queremos observar de la sucesión de Fibonacci.

Collatz:

Entrada: 23

Secuencia generada: 23 → 70 → 35 → 106 → 53 → 160 → 80 → 40 → 20 → 10 → 5 → 16 → 8 → 4 → 2 → 1

```
=== MENU DE OPERADORES RECURSIVOS ===  
1. Factorial  
2. Fibonacci  
3. Collatz  
4. Salir  
Elige una opcion: 3  
Ingresa un numero inicial para Collatz: 23  
Secuencia de Collatz: 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
```

Al elegir la opción 3 se despliega la opción para ingresar un numero inicial para que muestre una secuencia de Collatz a partir de ese número.

Salida del programa:

```
=== MENU DE OPERADORES RECURSIVOS ===  
1. Factorial  
2. Fibonacci  
3. Collatz  
4. Salir  
Elige una opcion: 4  
Saliendo del programa...
```

Por ultimo, la opción 4 hace que programa se cierre, en este caso debido a que el menú fue hecho con un do while hace que mientras la opción sea diferente de 4, el programa se repetirá.

4. Resultados

```
4 public static void main(String[] args) {
5     Scanner scanner = new Scanner(System.in);
6     int opcion;
7     do {
8         System.out.println("\nElige una opción");
9         System.out.println("1) Conjetura de Collatz");
10        System.out.println("2) Número factorial");
11        System.out.println("3) Serie de Fibonacci");
12        System.out.println("4) Salir");
13        System.out.print("Seleccione una opción: ");
14        opcion = scanner.nextInt();
15        switch (opcion) {
16            case 1:
17                System.out.print("Ingrese un número entero positivo: ");
18                int c = scanner.nextInt();
19                System.out.println("Secuencia de Collatz para " + c + ":");
20                collatz(c);
21                break;
22            case 2:
23                System.out.print("Ingrese un número entero no negativo: ");
24                int n = scanner.nextInt();
25                long resultado = factorialRecursivo(n);
26                System.out.println(n + "! = " + resultado);
27                break;
28            case 3:
29                System.out.print("¿Cuántos términos de Fibonacci deseas? ");
30                int m = scanner.nextInt();
31
32                System.out.println("Serie de Fibonacci:");
33                for (int i = 0; i < m; i++) {
34                    System.out.print(fibonacciRecursivo(i) + " ");
35                }
36                break;
37            case 4:
38                System.out.println("¡Adios!");
39                break;
40            default:
41                System.out.println("Opción no válida.");
42        }
43    } while (opcion != 4);
44    scanner.close();
45 }
```

En esta sección se muestra la función principal **Main** donde se desarrolla el menú donde con un `switch` se deciden a que caso se lleva, dentro de los casos solo se piden los números necesarios, se hace la llamada a cada función respectivamente y se imprimen los resultados.

```
49 public static void collatz(int c) {
50     while (c != 1) {
51         System.out.print(c + "\t");
52         if (c % 2 == 0) {
53             c = c / 2;
54         } else {
55             c = 3 * c + 1;
56         }
57     }
58     System.out.println("1");
59 }
```

Aquí podemos ver la función de **Collatz** donde se muestra que haciendo solo el uso de un *if-else* simple, se puede convertir en número que sea en el número 1.

```

61     public static int factorialRecursivo(int n) {
62         if (n == 0 || n == 1) {
63             return 1;
64         } else {
65             return n * factorialRecursivo(n - 1);
66         }
67     }

```

Se muestra la función de **Factorial Recursivo**, en esta primero con el if se muestra el caso *base o de rompimiento* y con el else se muestra el caso de recursión que es el que constituye en general la función.

```

69     public static int fibonacciRecursivo(int n) {
70         if (n == 0) {
71             return 0;
72         } else if (n == 1) {
73             return 1;
74         } else {
75             return fibonacciRecursivo(n - 1) + fibonacciRecursivo(n - 2);
76         }
77     }

```

Por ultimo se tiene la función de **Fibonacci** donde el caso baso es cuando se llega al 0 o a 1, la recursividad trata de ir sumando los dos anteriores al número deseado.

5. Conclusiones

La práctica mostró, de manera bastante clara, cómo un conjunto de ideas que en el papel parecen simples definiciones matemáticas pueden transformarse en algoritmos funcionales dentro de un programa en Java. El factorial, la sucesión de Fibonacci y la conjetura de Collatz dejaron de ser meras expresiones enunciadas en clase para convertirse en ejemplos vivos de recursividad y control de flujo.

El menú implementado con *do-while* y *switch* no solo organizó el código, sino que también permitió que el usuario jugara con las diferentes opciones como si se tratara de un pequeño laboratorio interactivo. Con ello se evidencia que, aunque los algoritmos tienen una base teórica rigurosa, su puesta en práctica puede ser tan intuitiva como entretenida. Hasta la recursividad, que en teoría asusta, se vuelve manejable e incluso divertida cuando se la aplica de forma concreta.

Referencias

- [1] Factorial.mx. (2025) Número y función factorial. Accedido: 26-ago-2025. [Online]. Available: <https://factorial.mx/numero-funcion-factorial>
- [2] We Are Drew. (2022) ¿qué es la secuencia de fibonacci? Publicado: 22-jul-2022; Accedido: 26-ago-2025. [Online]. Available: <https://blog.wearedrew.co/concepts/que-es-la-secuencia-de-fibonacci>
- [3] V. M. Manero and A. J. P. Sáez. (2021) La conjetura de collatz: un problema «sencillo» que desafiará tu intuición. Publicado: 18-ene-2021; Actualizado: 18-ene-2021; Accedido: 26-ago-2025. [Online]. Available: https://www.abc.es/ciencia/abci-conjetura-collatz-problema-sencillo-desafiara-intuicion-202101180110_noticia.html